

RCX Tutorial

[Commands](#) [Sensor Watchers](#) [Stack](#)
[Controllers](#) [My Commands](#)



The following is a list of commands available to you for programming the robot (See advanced below)

On



Turns motors (connected to ports A, B, or C) on (moves robot forward or backwards).

On For



Allows robot to move for a specific time period (in seconds).

Off



Turns motors off (stops robot).

Wait (does not mean "stop & wait," only wait)



Does not mean stop! Instead causes robot to wait a specific period of time before executing the next command.

Set Direction



Sets the direction that the motors turn. Need to download a sample program to determine which direction is forward and which is backwards, as it depends on the orientation of the wire connection.

Reverse Direction



Changes the direction of the motors.

Beep



Causes robot to beep a specific number of times. Good debugging tool.

Tone (music)



Similar to beep, but plays a different tone for a specific period of time. Can be used to write music!

Power



This command sets the power level (1 too low, 8 max).

Advanced Programming Commands

Counter Variable



These commands can be used to keep track of the number of times certain events occur (for example how many times the touch sensor is pressed or the robot has turned). At the beginning of the program, the counter always = 0. To use, simply place the "Add to

Counter" button on top of the command(s) you want to count. It is a good idea to reset the counter when you are done using it (so if later in the program, the robot needs to count something else, it will start from 0).

Reset Message



When receiving a message from another RCX, this command allows you to reset it.

Reset Timer



A timer can be used for a number of tasks. Be sure to reset it when you are done.

[BACK](#)



For all sensor watches, you can create two lists of program commands, depending on the state of the sensor. Note - you don't have to write code for both conditions. If one is left blank, then the program will revert to the main program.

Touch Sensor



Select the port (1-3) that the touch sensor is attached. Usually it is easiest in a complicated program to only write code for the "Press" condition.

Light Sensor



Use the view button on your RCX to determine the light intensity in the location your robot will be running. Use these values to set the range for dark and bright.

Timer



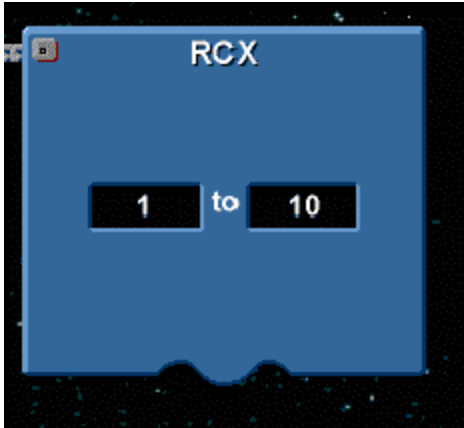
A timer can be used to execute a certain amount of code during a given time or to start executing part of the code after a specific period of time has passed.

Counter



The timer and counter can be used to program the robot to do certain tasks after a period of time or after a specific set of events occurs.

RCX Signals



This command allows you to program your robot to do certain commands when it receives a signal. For example, if the numbers are the same "1 to 1", then the program will execute Message 1.

[BACK](#)



Stack Controllers allow you to create programs for robots that are versatile. These commands represent "repetition" and "conditional" statements.

Repeat (# of times)



To repeat a set of commands, simply place them in between the "begin" and "end" and set the number of repetitions.

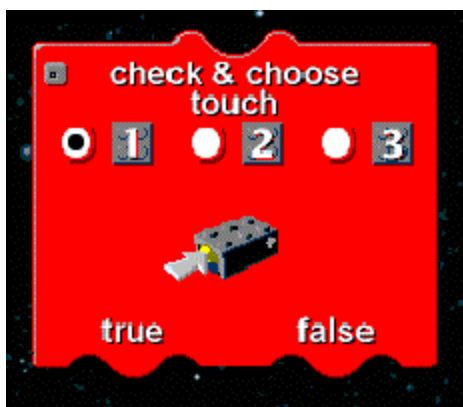
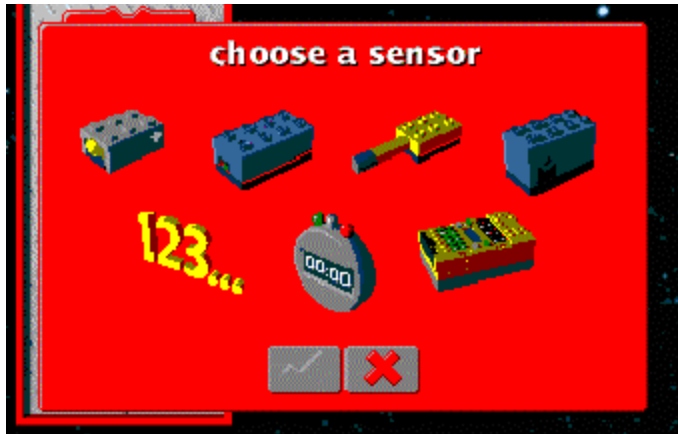
Repeat Forever



Why would you ever want something to repeat forever? If it is part of a conditional command, then it will repeat as long as the condition is true, and stop once the condition is no longer true. For example, you could place this command on the "Release" end of the Touch Sensor command.

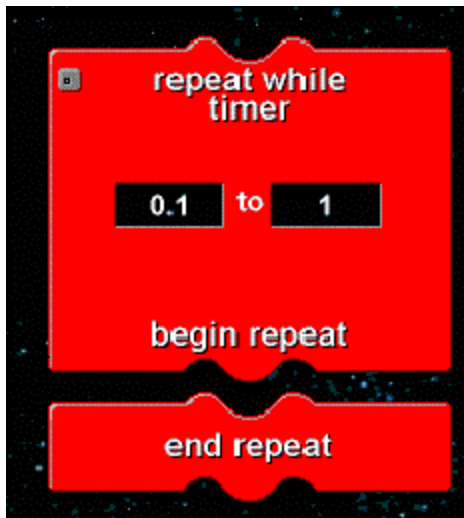
Check & Choose

For "Check and Choose" Sensor Watcher, it is necessary to choose a sensor before programming the code.



Here is an example of a "Check and Choose" for a touch sensor. (This is very similar to "If then" statements if you have programmed before). It is programmed similar to the other sensors, with true and false conditions.

Repeat while



Here is an example of "Repeat While" command. This command executes as long as the timer values are within a certain range and stops (exits the loop) once the timer exceeds the final value.

Wait Until



This command allows the robot to "Wait Until" the light sensor reaches a specific value before executing a command.

[BACK](#)



"My Commands" allow you to create subroutines for your programs. Subroutines are essentially small parts of programs. The benefits of using subroutines include the following:

- makes your programs easier to read,
- saves time since you only need to create the code once to use over and over
- allows multiple people to work on the same project (industry)

"My Commands" Structure



When using "My Commands" in RCX programming, you will need to give them a name, and also plug in commands between the "begin" and "end"

Example #1



For example, here is a subroutine that causes the robot to drive backwards.

Example #2



Lastly, you can minimize subroutines, so they take up less space. You can also stack subroutines just like regular commands.

[BACK](#)

